2006 World Congress on Computational Mechanics
Mini-Symposium on Computational Mechanics with Adaptive Mesh Refinement

Applying Parallel Adaptive Methods with GeoFEST/PYRAMID to Simulate Earth Surface Crustal Dynamics

Charles D. Norton

Greg Lyzenga, Jay Parker, Margaret Glasscoe, Andrea Donnellan and Peggy Li

Jet Propulsion Laboratory

California Institute of Technology

July 21, 2006

Los Angeles, California

Why Adaptive Mesh Refinement?

- **AMR simultaneously improves** solution quality, time to solution, and computer memory requirements when compared to generating/running on a globally fine mesh.

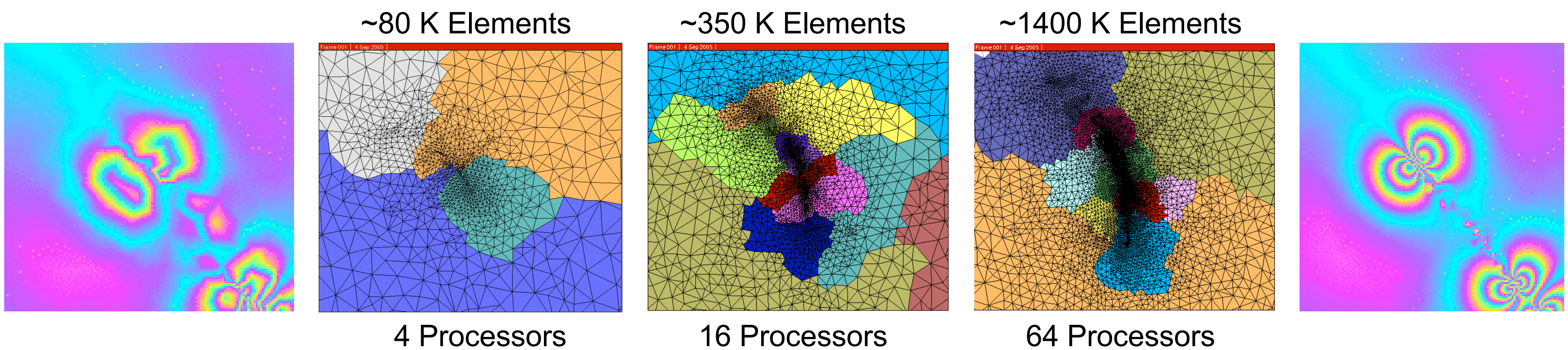
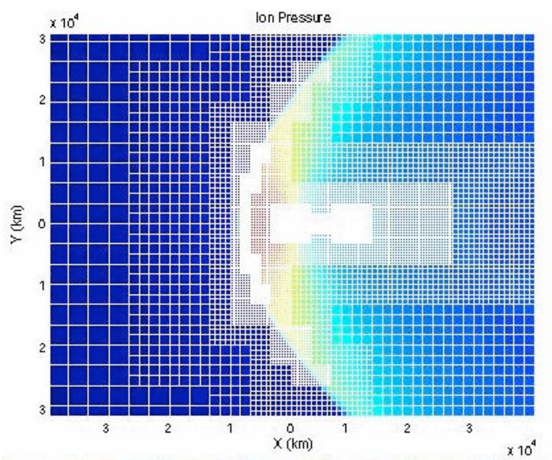


Illustration of AMR showing improvement in surface displacement solution quality with mesh density (Landers faulted mesh solved with GeoFEST/Pyramid).

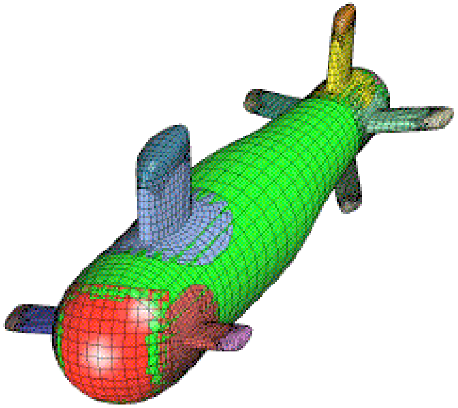
- **Future proposed NASA missions**, such as InSAR for Earth surface deformation and other measurements, will require support for large-scale adaptive numerical methods using AMR to model observations.
- **AMR is applied across disciplines** and various mesh geometries, but has seen the greatest application and success in computation fluid dynamics for predictive simulation of complex flows around complex structures.

What is the State of the Art for Parallel AMR?

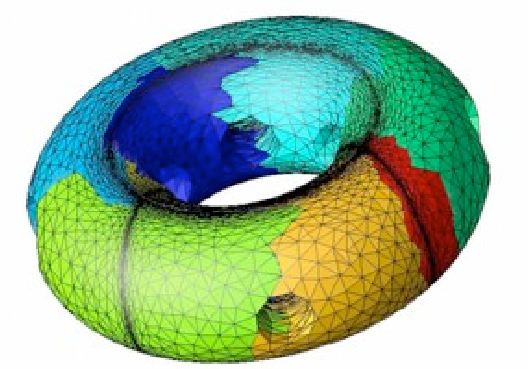
- **Structured AMR** is well established for rectilinear block-adaptive geometries.
- **Unstructured AMR** has been applied both for arbitrary geometries and where structured methods have also be used.
- **Composite/Overlapping AMR** allows “patching” of structured grids to gain benefits of structured approaches for CFD problems (conservation) while also supporting complex geometries of unstructured methods



Structured AMR: Cometary atmospheric pressure field using PARAMESH. M. Benna et. al., GSFC.



Composite AMR: Submarine mesh gridded using Overture. W. Henshaw, et. al. LLNL



Unstructured AMR: Torus with 4 holes gridded using FMDB. E. Seol and M. Shepard, Rensselaer.

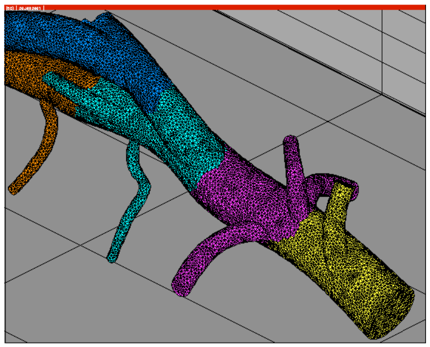
What Tools Support Parallel AMR?

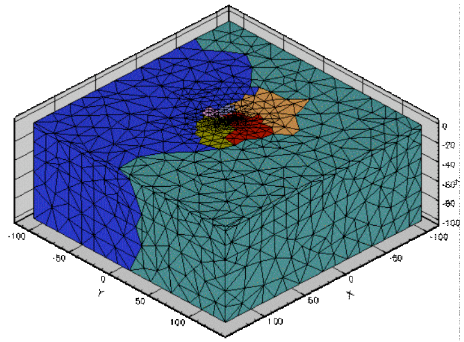
Tool Name	Category	Comments
CART3D [Aftosmis] **CHOMBO** [Coella] **PARAMESH** [MacNeice] **BATS-R-US** [Hansen/Gombosi] Many Other Tools Exist…	Structured AMR	World-Class tools. Only PARAMESH supports Fortran-based solvers directly. Complex interfaces exist.
Overture [Henshaw]	Composite AMR	Successful, but mesh generation complexity has limited usage by ordinary users.
SUMMA3D [Freitag] **PAOMD** [Remacle/Shephard] Many CAD tools exist for mesh generation, but are not designed for AMR and integration with solvers	Unstructured AMR	Very limited category. Mesh smoothing is generally used to control element quality. Only C++/C implementations available with complex interfaces to solvers.
PYRAMID [Lou/Norton]		Fortran-based library (with C adapters) and simple interface.

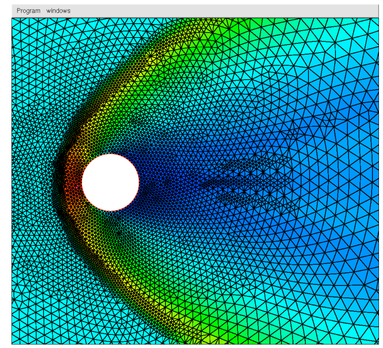
- **Many good sequential tools exist for mesh generation/solution with AMR**

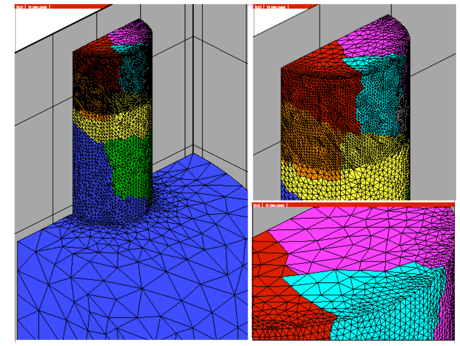
Overview of Pyramid Parallel AMR Library

Modern... Simple... Efficient... Scalable...









Task Objective

Development of a Fortran object-based software library supporting parallel unstructured adaptive mesh refinement for large-scale scientific & engineering modeling applications.

Design Approach

- Efficient object-based design in Fortran 90/95 and MPI.
- Automatic mesh quality control, dynamic load balancing, mesh migration, partitioning, integrated mathematics and data accessibility routines, easy solver integration.
- Scalable to hundreds of processors and millions of elements using triangles (2D) and tetrahedra (3D).
- Ease of use with development driven by application needs.
- Only refinement is officially supported at this time, but a experimental coarsening capability exists

NASA Programmatic Relevance

- Large scale modeling and simulation applications with complex geometry including support of ESTO/CT Round III teams such as Solid Earth tectonics modeling and more.

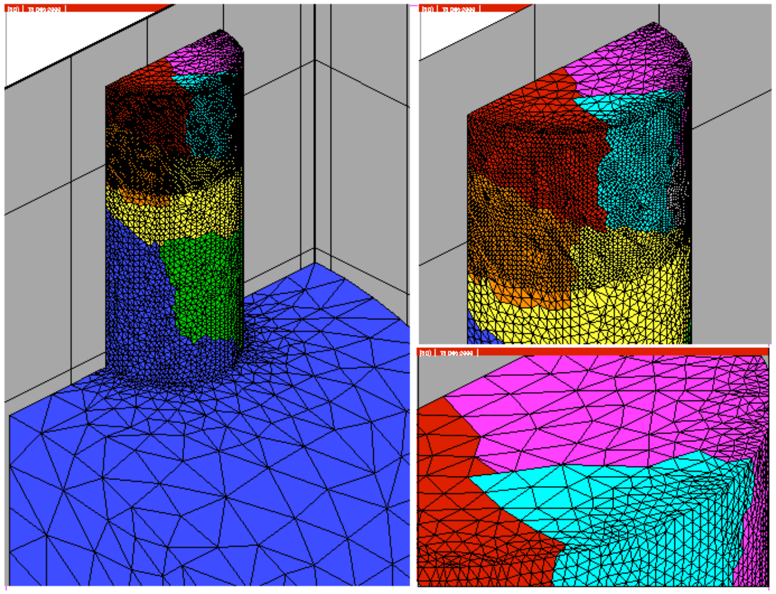
Relevant Application Areas

- Structural modeling and engineering mechanics for Earth and Space science applications.
- Fluid mechanics and gas dynamics.
- Solid Earth active tectonics simulation models.
- Design modeling of microwave active devices.
- Fast mesh generation from high quality coarse meshes.

Charles D. Norton, John Z. Lou, Thomas A. Cwik, and E. Robert Tisdale

Benefits of AMR vs. Software Complexity

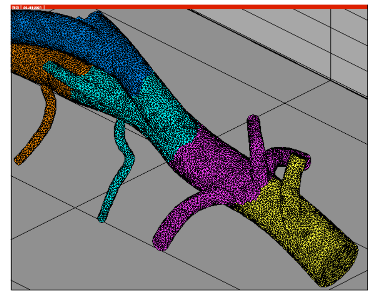
- **Fewest DOF (nodes) required** for a problem solution of a given accuracy.
- **Saves orders of magnitude in CPU and memory requirements** for some classes of problems enabling previously intractable problems.
- **Required in Earth/Space Sciences** for problems with large spatial/temporal dimensions.
- **Avoids trial-and-error in grid selection** especially for problems with traveling discontinuities.
- **Unstructured AMR captures complex geometry,** but structured AMR software design is easier.
- **Maintaining good element geometry is not a solved problem** when considering trade-offs in mesh interpolation and element creation.

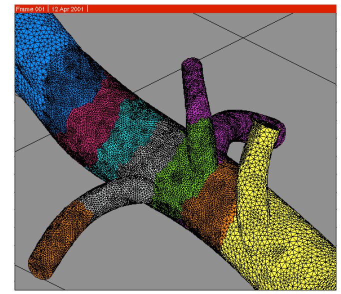


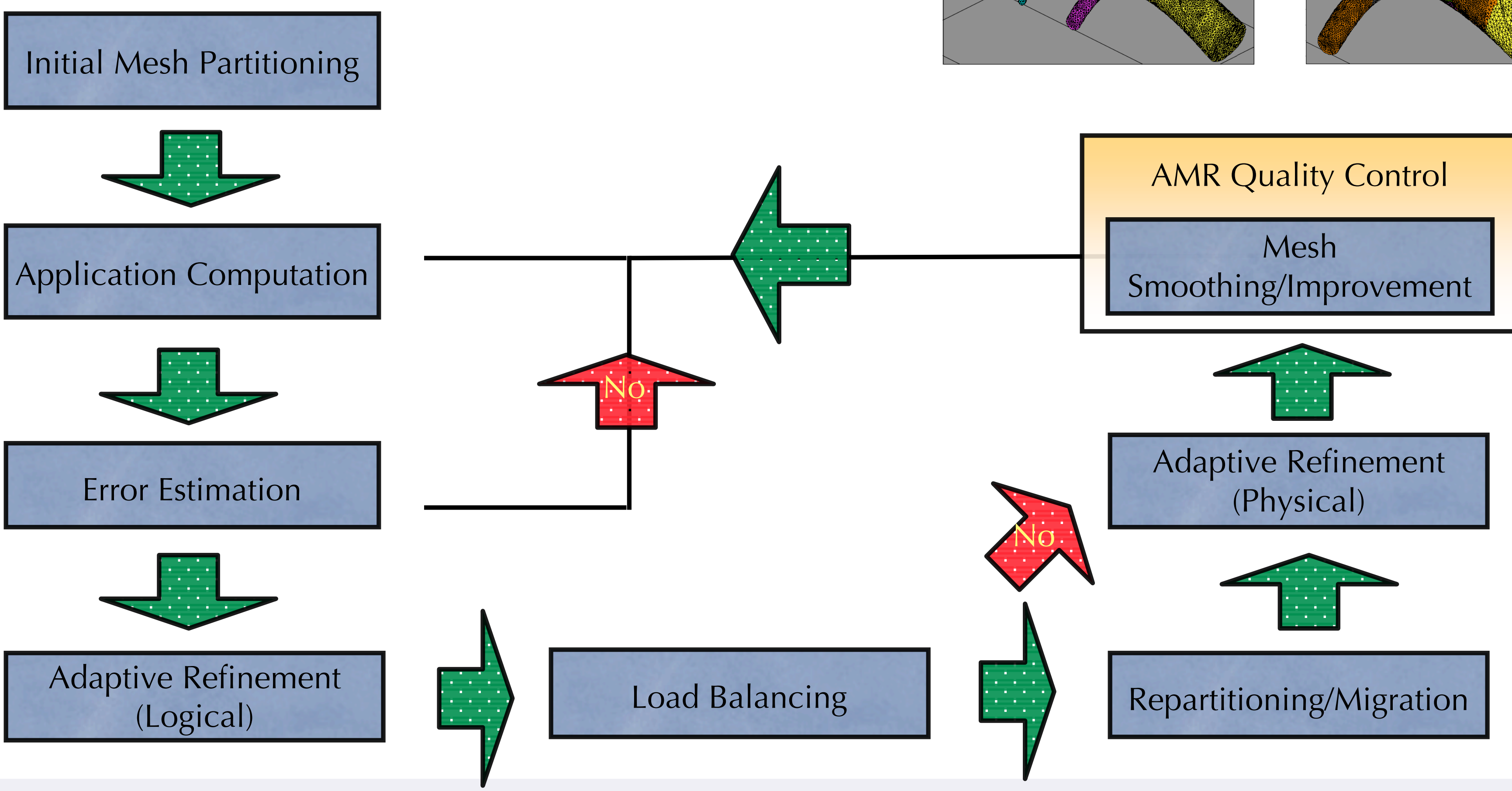
- **Very few general purpose packages** and almost none are Fortran-based.
- **Error estimation can be complex** when determining criteria to drive refinement.

Pyramid Parallel AMR Process

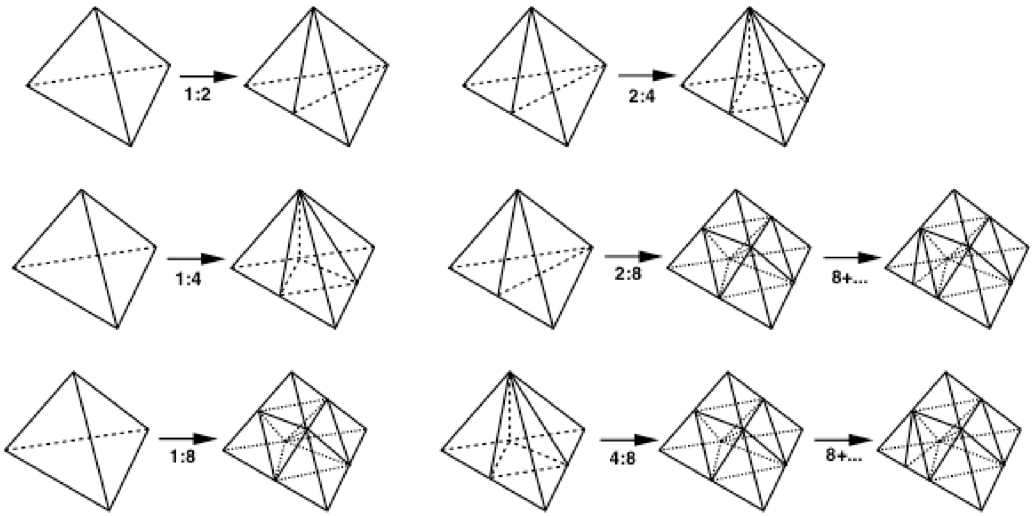
- **Partitioning, Load Balancing, Adaptive Refinement, Mesh Migration, Quality Control**



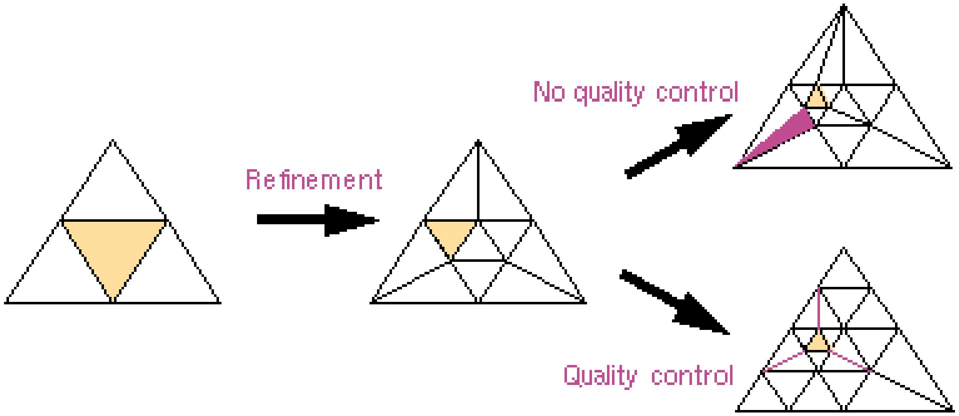




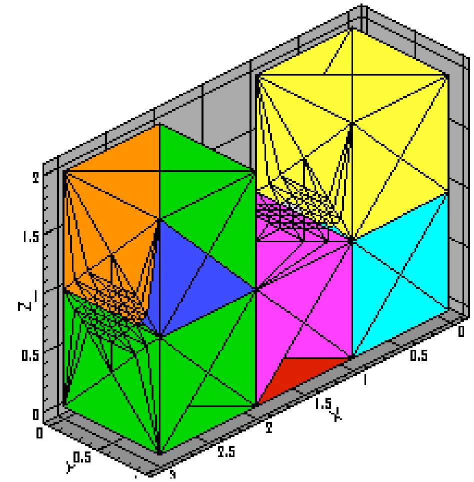
Element Refinement and Coarsening



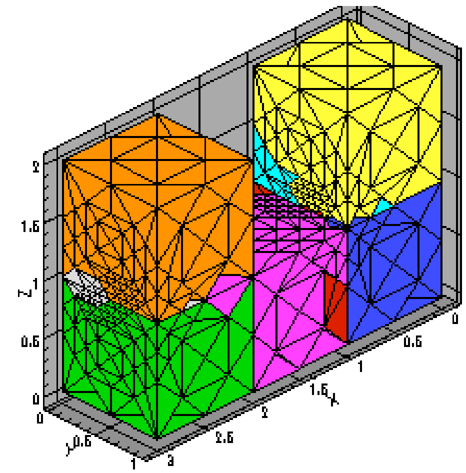
Refinement Patterns help ensure mesh consistency (no hanging nodes)



Quality Control Algorithm shown in 2D



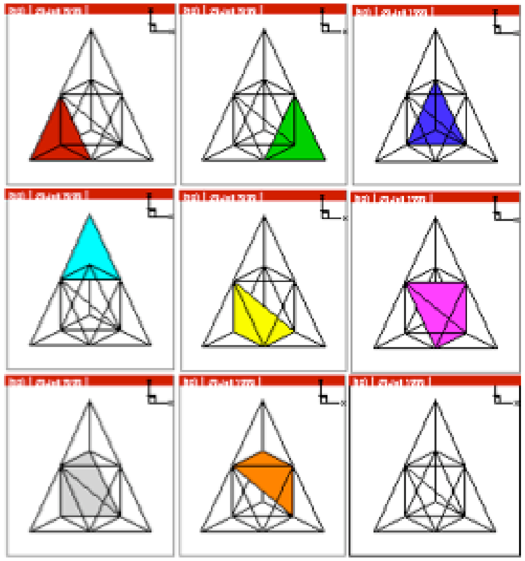
No Quality Control gives poor aspect ratios

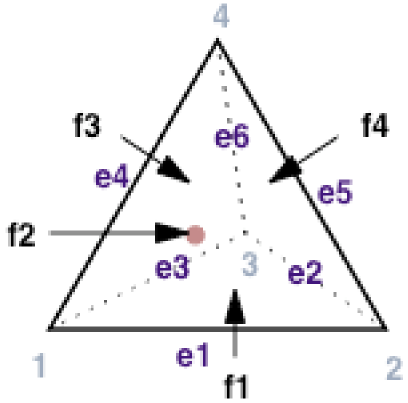


Quality Control maintains element geometry better

Implementation Issues for Parallel AMR and GeoFEST

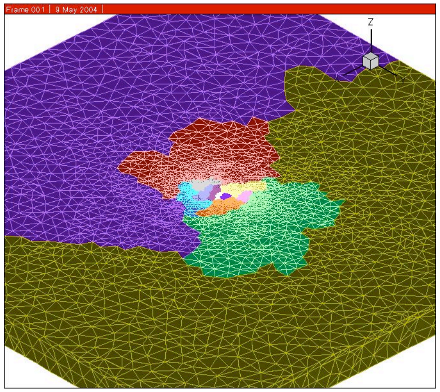
- **Element labeling schemes and per-processor indexing** should be consistent across codes.
- **Mesh creation and conversion is required** as pre-processing steps.
- **Pyramid only supports tetrahedral elements** for parallel AMR (currently).
- **Two representations of mesh geometry** are needed as GeoFEST maintains an internal version of the mesh (simplifies interfacing, but adds storage overhead).
- **AMR refinement overhead is small** even for meshes with 10s of millions of elements where 100s of millions can be supported.
- **Visualization of large data runs is complex** and often requires use of parallel visualization/animation tools like RIVA.

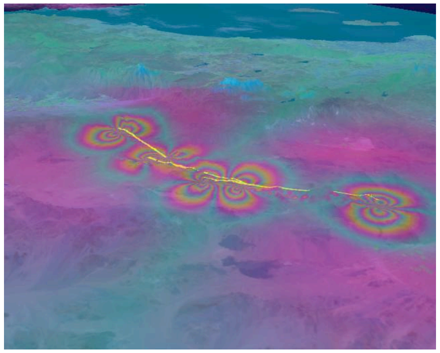


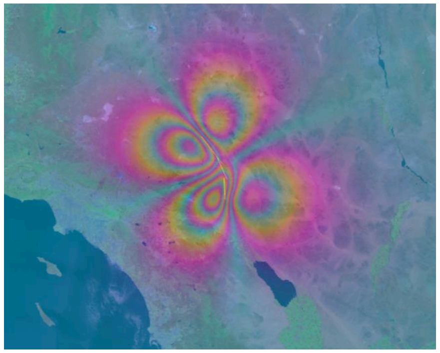


Overview of Using AMR in GeoFEST

- **GeoFEST performs operations to modify the "state" of Pyramid objects and relies on Pyramid for information and operations related to parallel processing**
- **PreProcessing**
  - Run GFMeshParse on GeoFEST input to create Pyramid input
- **Processing**
  - Specify # of field variables to track on each mesh component
  - Distribute GeoFEST input data using Pyramid mesh partitioning
  - Solve on the coarse mesh
  - Apply per-element error estimation criteria (strain energy)
  - Perform AMR (logical refinement, load balancing, physical refinement)
  - Update GeoFEST data structures and interpolate to new field variables
  - Solve on the refined mesh
- **Post Processing**
  - Visualize results by transformations to TecPlot format





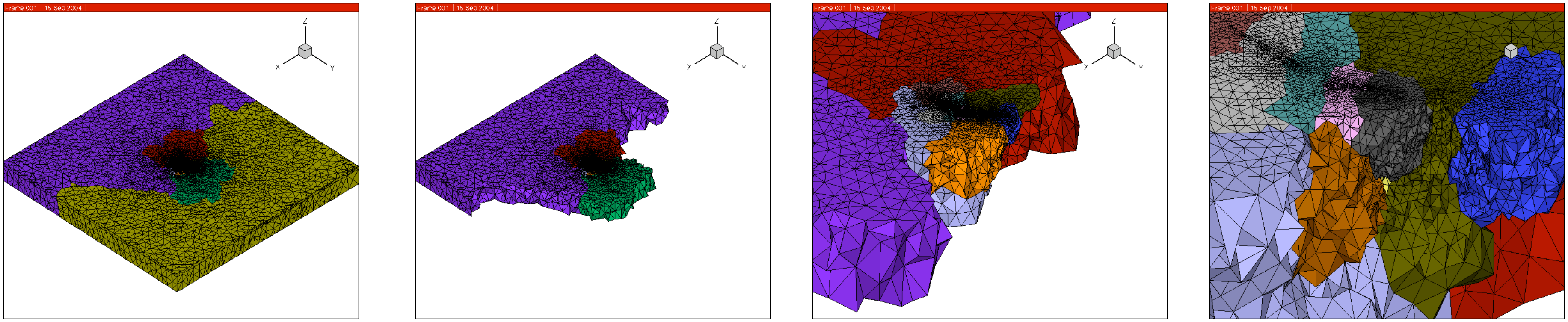


Sample Pyramid Command

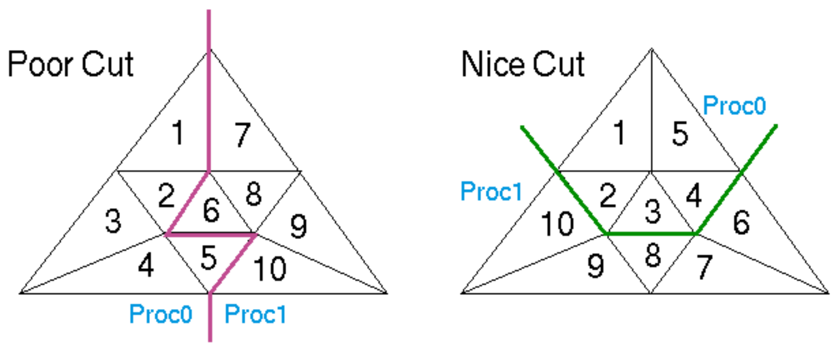
- **PAMR_GET_PARTITION_NODES**
- **Interface**
  - function PAMR_GET_PARTITION_NODES(this) RESULT(this)
- **Arguments**
  - type (mesh), intent(in) :: this
  - Integer, dimension(this%loc_boundary_nodes) :: terms
- **Description**
  - Returns, in a one-dimensional array, the global_ids of nodes on the partition boundary.
- **Notices**
  - Use PAMR_NODE_PARTITION_COUNT to get the size of the array to allocate for the result of this function.

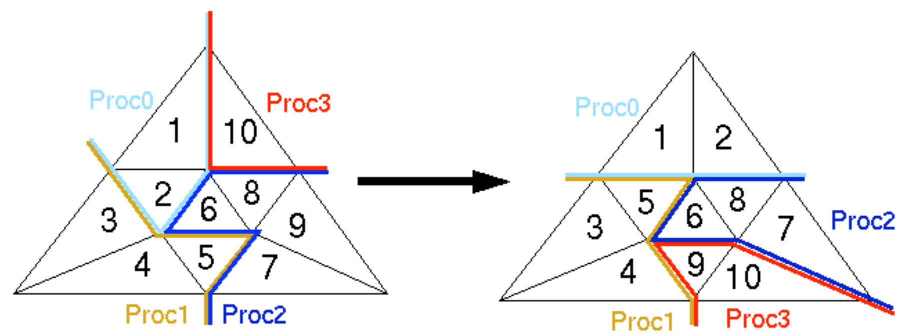
- **Most Pyramid Commands** operate on a “mesh” type/object as the primary mechanism to modify and or access mesh components.
- **Mesh index locations** can be returned as well to map directly into GeoFEST storage.
- **Over 100 Commands Exist** but really only a handful are needed to be productive.

Mesh Partitioning, Migration, & Load Balancing



Element-based Partitioning achieves balanced load, connected components, and reasonable edge-cut quality. Coarse elements are weighted, migrated, then refined to load-balanced partitions.

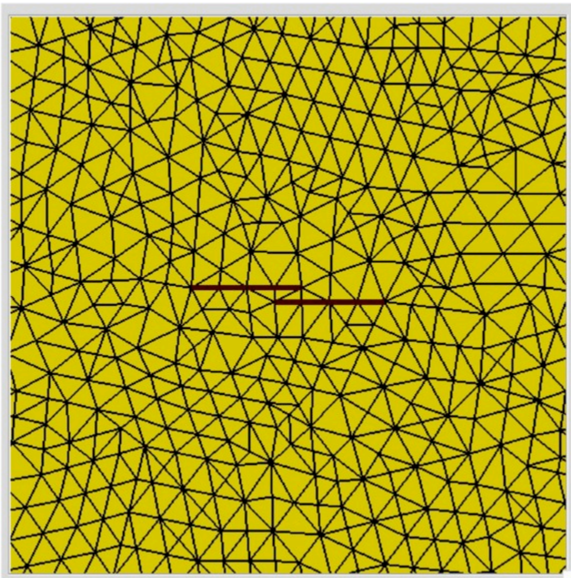




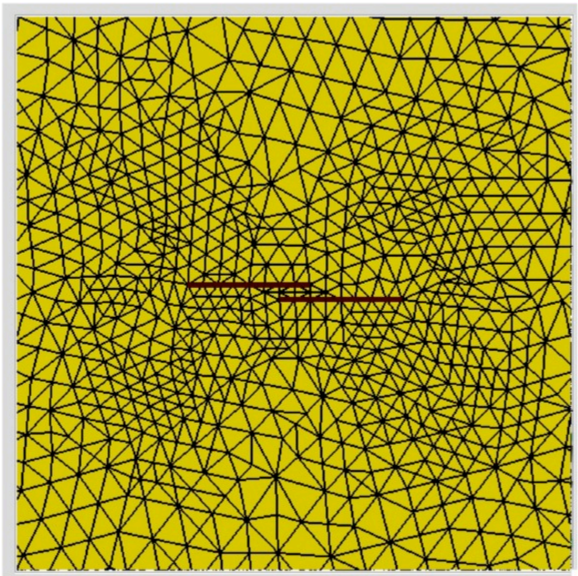
Graph Partitioning represents an important aspect of minimizing communication at boundaries.

Mesh Migration Algorithms must handle irregular redistribution of mesh components efficiently and correctly

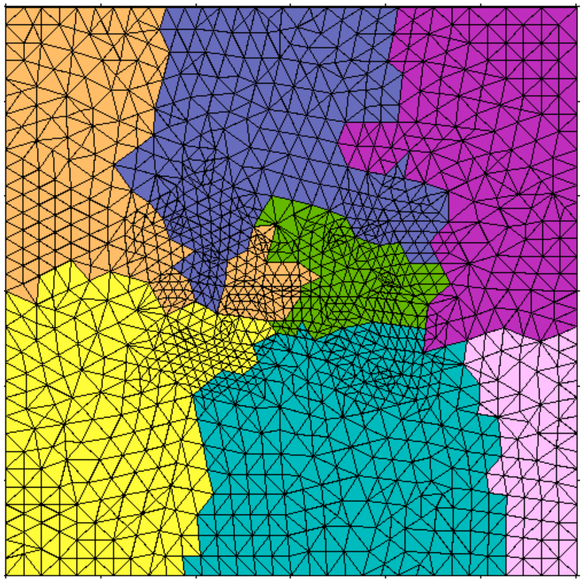
Solution Driven AMR for Fault Stepover



Initial Coarse Stepover Mesh (2D Surface of 3D Mesh Shown)



Yee-Refinement with Mesh Smoothing



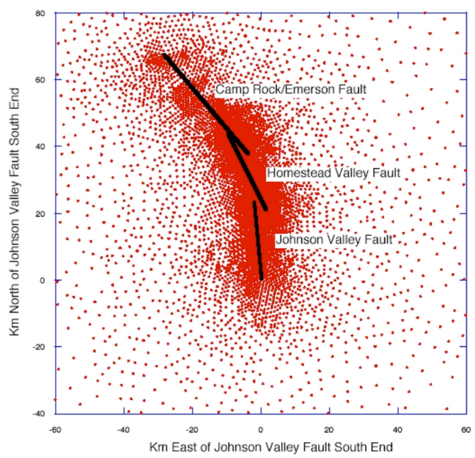
Pyramid Refinement with Mesh Quality Control

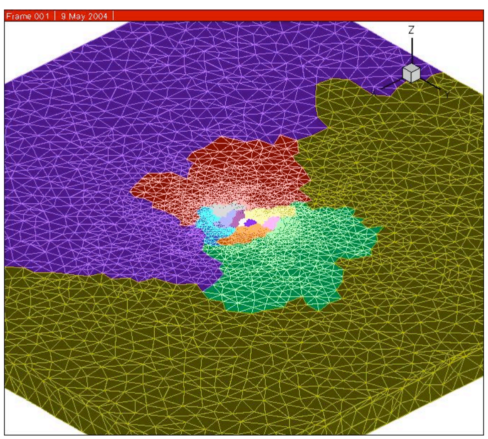
Strain-Energy-Based Refinement where additional nodes are added by two different AMR processes. Yee-refinement scheme "smooths" node positions to maintain reasonable element aspect ratios. PYRAMID AMR scheme uses quality-control maintaining coarse mesh node positions

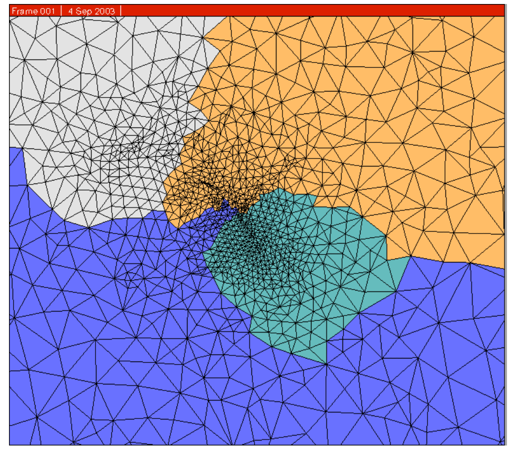
Solution Interpolations may be slightly less precise when mesh smoothing is performed, however, if refinement/coarsening is applied the quality of the Pyramid solution is driven by the quality of the initial coarse mesh (a trade-off)

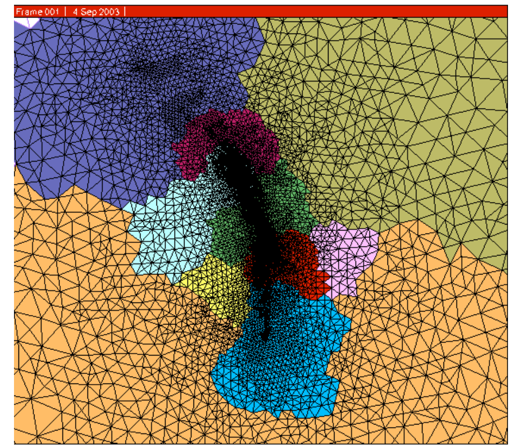
Direct Mesh Generation for Landers Simulation

Improvement in Fringes illustrated when AMR is applied

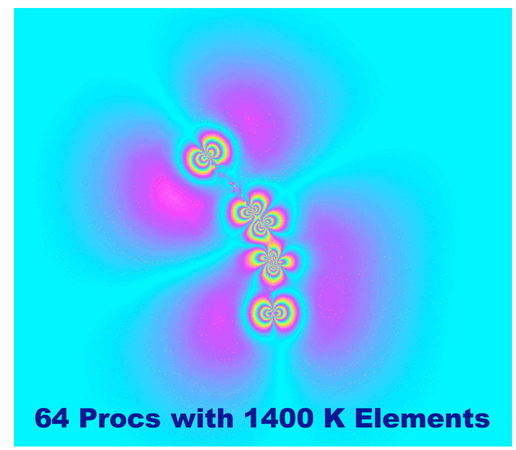








4 Procs with 82 K Elements

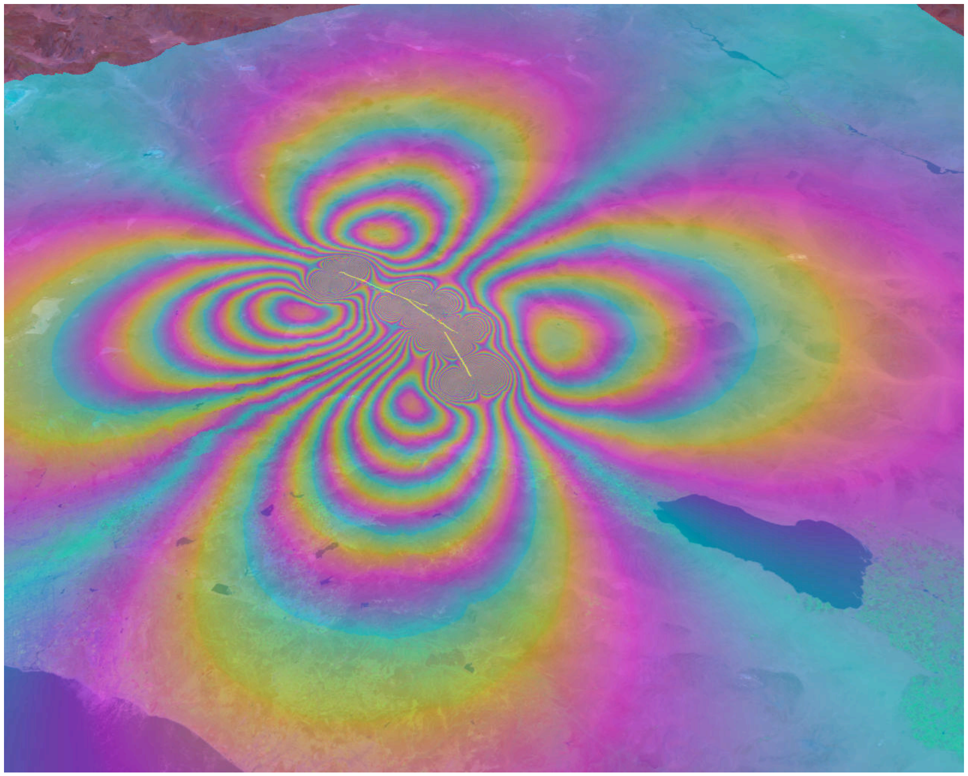


64 Procs with 1400 K Elements

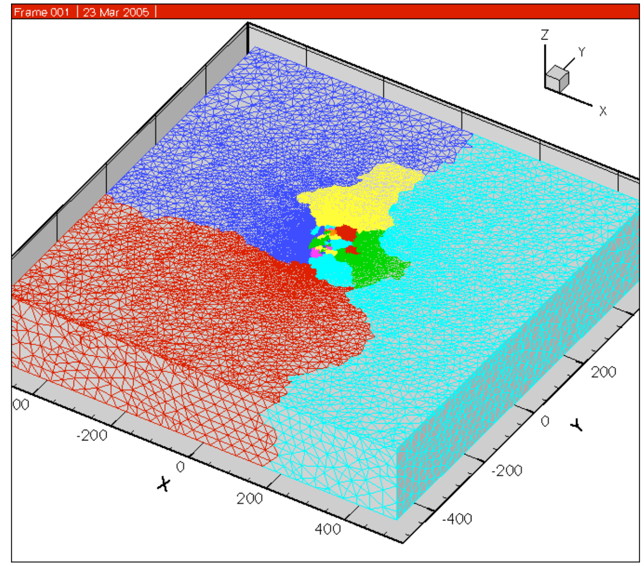
Interferegrams of Elastic Displacement where direct addition of nodes around the fault region, with parallel mesh partitioning, improves resolution.

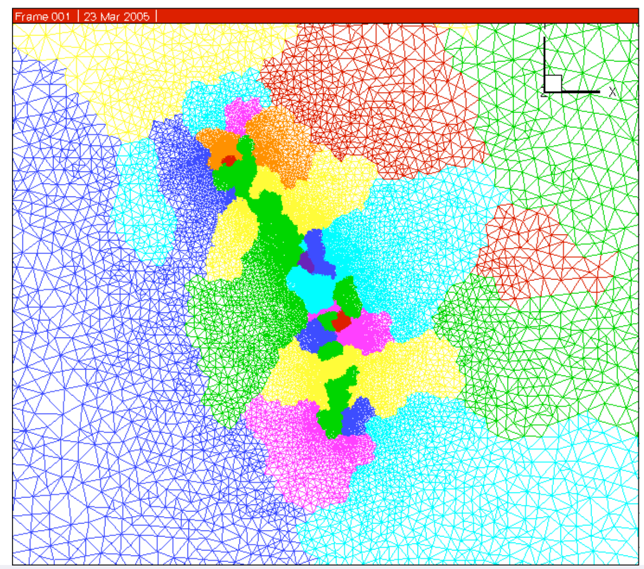
Solution-Driven AMR for Landers Simulation

Views of Parallel AMR applied to form ~16 M element mesh that could not be generated using GuiVisco sequentially

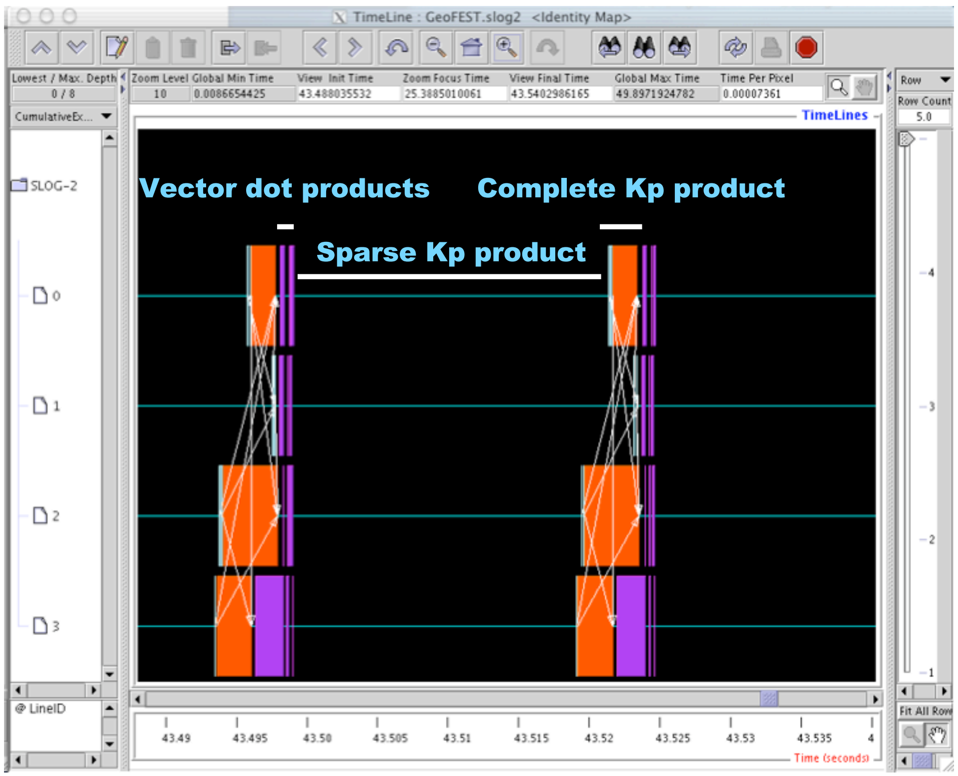


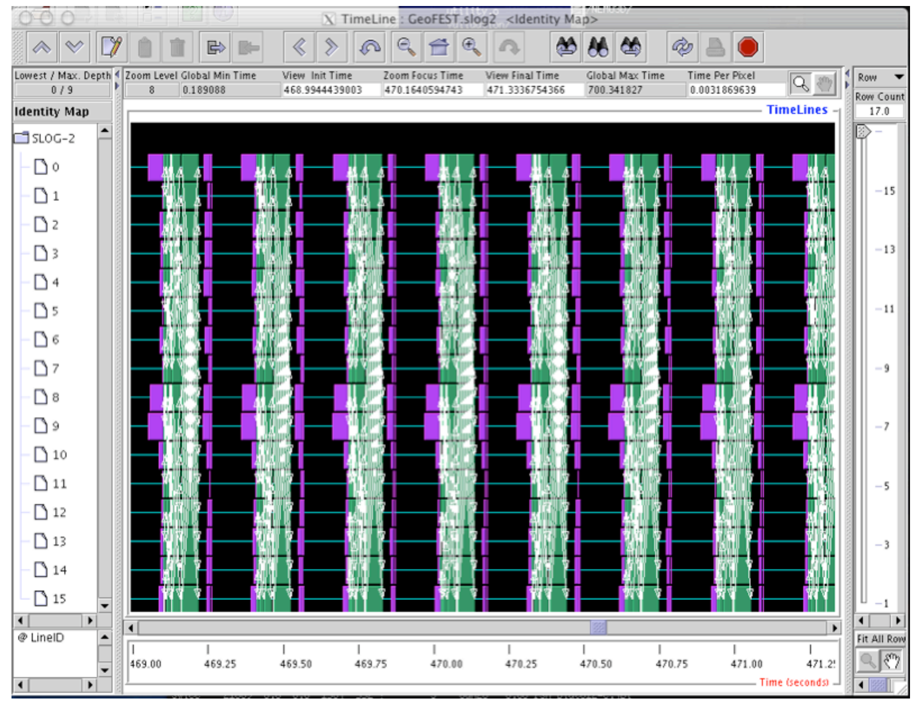
GeoFEST Simulated Surface Displacement from coseismic Landers model. Viscoelastic phase (not shown) run for 500 year simulation on ~500 procs over ~12 hours where AMR processing is negligible.





Communication Performance and Optimization





Non-Scalable Solution

Two Iterations of PCG solve of Thousands Performed show communication efficiency for a 4 processor example. Black shows computation, Red shows WAITALL (completion of matrix-vector products) and violet ALL_REDUCE (global combine of parts of vector dot product) communication. Computation dominates giving scalability.

Improper Communication Scheme focuses on balanced communication that does not scale since more communication operations are applied than is strictly needed.